

## Κεφάλαιο 1. Κάθε αρχή και...εύκολη

### 1.1 Σκοπός του βιβλίου

Καθημερινά χρησιμοποιούμε διάφορες υπολογιστικές εφαρμογές, όπως κειμενογράφους για να γράφουμε τα κείμενά μας και διαδικτυακούς περιηγητές για να εξερευνούμε το διαδίκτυο. Έχετε αναρωτηθεί πως δημιουργούνται οι εφαρμογές αυτές; Θα θέλατε να δημιουργήσετε τις δικές σας εφαρμογές; Σκοπός αυτού του βιβλίου είναι να σας εισάγει στον κόσμο του προγραμματισμού, ένα κόσμο που θα μας επιτρέψει να δημιουργούμε τα δικά μας προγράμματα με ένα, όμως, ιδιαίτερος διασκεδαστικό και δημιουργικό τρόπο. Πως θα σας φαινόταν αν αντί για απλοί χρήστες ενός παιχνιδιού μπορούσατε να γίνετε οι κατασκευαστές του; Θα θέλατε να μπορείτε να προσδιορίσετε τη συμπεριφορά των ηρώων σας, τότε πυροβολούν, τότε χάνουν, ποια εμπόδια πρέπει να αποφύγουν, ποιους εχθρούς μπορούν να εξολοθρεύσουν ή πόσα νομίσματα πρέπει να συλλέξουν για να κερδίσουν;



Θα συζητήσουμε για όλα αυτά μέσα από τη χρήση της εφαρμογής προγραμματισμού MSKodu, ενός περιβάλλοντος προγραμματισμού που προσφέρετε δωρεάν από τα ερευνητικά εργαστήρια της Microsoft. Ένα περιβάλλον που δημιουργήθηκε για να εμπνεύσει μικρούς και μεγάλους ώστε να δημιουργήσουν εύκολα και γρήγορα συναρπαστικά παιχνίδια! Ο κεντρικός ήρωας, που θα κληθείτε να οδηγήσετε σε συναρπαστικές περιπέτειες, είναι ο Kodu που φαίνεται στη διπλανή εικόνα.

Στο τέλος της μελέτης του βιβλίου μας, θα είστε σε θέση να προγραμματίζετε τον ήρωα μας ώστε με τη βοήθεια του χρήστη να ανατινάξει κάστρα, να πυροβολεί τους αντιπάλους του, να τρώει μήλα, να κολυμπά, να κερδίζει πόντους και ζωές. Θα μπορείτε να φτιάξετε ένα παιχνίδι στο οποίο ο παίκτης θα παίζει ποδόσφαιρο σε ένα γήπεδο και θα προσπαθεί να σκοράρει ή ακόμη θα προσπαθεί να σώσει την αγαπημένη του Kodula περνώντας μέσα από χαράδρες με λάβα και ψηλά βουνά.



Καλωσορίσατε στον μαγικό κόσμο του προγραμματισμού με το MSKodu! Σας ευχόμαστε καλή διασκέδαση!

### 1.2 Πρόγραμμα και προγραμματισμός

Τι είναι όμως το πρόγραμμα; Ας δούμε πως το ορίζει η βικιπαίδεια:

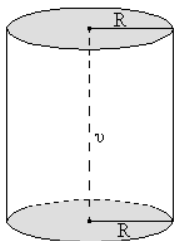
*Στην επιστήμη υπολογιστών με τον όρο πρόγραμμα αναφερόμαστε σε μια συγκεκριμένη ακολουθία εντολών τις οποίες πρέπει να εκτελέσει ένας υπολογιστής για να παραγάγει το επιθυμητό για το χρήστη αποτέλεσμα.*

Σας μπέρδισε; Ας κάνουμε τα πράγματα πιο απλά: **Πρόγραμμα** είναι η ακολουθία των βημάτων που πρέπει να γίνουν για να λυθεί ένα πρόβλημα. Δυο είναι επομένως τα βασικά μας στοιχεία, **το πρόβλημα** και **ο τρόπος επίλυσής του**.

Τι προβλήματα συναντάμε στον κόσμο των υπολογιστών; Θέλουμε να παίζουμε παιχνίδια, να γράφουμε κείμενα, να επισκεπτόμαστε ιστοσελίδες στο διαδίκτυο, να κάνουμε σύνθετες πράξεις για να προβλέψουμε τον καιρό. **Προγραμματισμός**, λοιπόν, είναι η διαδικασία με την οποία προσπαθούμε να δώσουμε τις κατάλληλες οδηγίες στον υπολογιστή για το πώς θα λύσει τα συγκεκριμένα προβλήματα ενώ **προγράμματα** είναι τα τελικά σύνολα των εντολών που εμείς ως προγραμματιστές καταλήξαμε ότι λύνουν τα προβλήματά μας.

Όμως ο προγραμματισμός είναι μια γενική έννοια που δεν συναντάμε μόνο στους υπολογιστές αλλά και στην καθημερινή μας ζωή. Δημιουργούμε προγράμματα για να λύνουμε προβλήματα της καθημερινότητά μας.

**ΚΥΛΙΝΔΡΟΣ**



Εσείς στο σχολείο καλείστε κυρίως να λύσετε μαθηματικά προβλήματα που περιλαμβάνουν υπολογισμούς και λογικές πράξεις, όπως π.χ. ο υπολογισμός του όγκου ενός κυλίνδρου. Πώς θα επιλύατε ένα αντίστοιχο πρόβλημα; Θα ακολουθούσατε μια διαδικασία δυο βημάτων.

Το πρώτο βήμα θα ήταν να υπολογίσετε το εμβαδόν της βάσης του κυλίνδρου που βρίσκεται από τον τύπο

$$\pi \cdot r^2$$

Το δεύτερο βήμα σας θα ήταν να πολλαπλασιάσετε το εμβαδόν που βρήκατε με το ύψος του κυλίνδρου ώστε να έχετε ως αποτέλεσμα τον όγκο του κυλίνδρου.

$$V = E_{\beta} \cdot u = \pi R^2 u$$

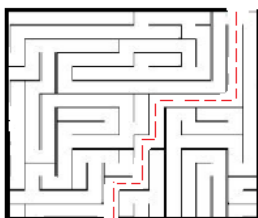
Ένα άλλο πρόβλημα είναι και ο τρόπος που θα μπει μια τάξη 130 μαθητών σε λεωφορεία χωρητικότητας 50 ατόμων για να πάει εκδρομή με το σχολείο. Πως θα λύνατε αυτό το πρόβλημα; Θα ξεκινούσαμε υπολογίζοντας ποσά λεωφορεία θα χρειαστούν. Αφού το κάθε λεωφορείο χωράει 50 μαθητές και η τάξη έχει 130 μαθητές θα χρειαστούν 3 λεωφορεία για να σας μεταφέρουν. Δεύτερο βήμα μας θα είναι να βάλουμε τους μαθητές ισομερώς στα λεωφορεία. Έτσι διαιρούμε τον αριθμό των μαθητών με τον αριθμό των λεωφορείων που υπολογίσαμε στο πρώτο βήμα. Επειδή προκύπτει κάτι πάνω από 43, βάζουμε τους 43 πρώτους μαθητές στο πρώτο λεωφορείο. Έπειτα βάζουμε τους επόμενους 43 στο δεύτερο λεωφορείο και τέλος τους τελευταίους 44 μαθητές που περισσεύουν στο τρίτο λεωφορείο.



Για να λύσουμε τα παραπάνω προβλήματα ακολουθήσαμε μια σειρά από βήματα. Ο τρόπος που σκεπτόμαστε για να λύσουμε ένα πρόβλημα ονομάζεται αλγόριθμος. Κάθε πρόγραμμα “κρύβει” από πίσω του έναν αλγόριθμο. Όπως ίσως ήδη έχετε αντιληφθεί για να λύσουμε το πρόβλημα και να χτίσουμε σωστά τον αλγόριθμο πρέπει να κατανοήσουμε πλήρως το πρόβλημα, το περιβάλλον του και τις προϋποθέσεις που μας δίνονται.



Για να δούμε ακόμη ένα πρόβλημα, το πρόβλημα του λαβυρίνθου. Το ζητούμενο είναι ο ιππότης να βγει από τον λαβύρινθο και να σώσει την πριγκίπισσα. Για να λύσουμε το πρόβλημα έχουμε ως διαθέσιμες οδηγίες τις επιτρεπτές κινήσεις που μπορεί να κάνει ο ιππότης. Μπορεί να κινηθεί κατά ένα βήμα δεξιά, αριστερά, πάνω και κάτω αλλά δεν μπορεί να διαπερνά τους τοίχους. Πώς θα καθοδηγούσαμε λοιπόν τον ιππότη για να σώσει την αγαπημένη του;



Ένα σύνολο οδηγιών που θα έλυνε επιτυχώς το πρόβλημα είναι το ακόλουθο:

Ιππότη

1. Κάνε πέντε βήματα προς τα κάτω
2. Κάνε τέσσερα βήματα προς τα δεξιά
3. Κάνε τέσσερα βήματα προς τα δεξιά
4. Κάνε ένα βήμα προς τα δεξιά
5. Κάνε δύο βήματα προς τα κάτω
6. Κάνε δύο βήματα προς τα δεξιά
7. Κάνε δύο βήματα προς τα κάτω

Να η πριγκίπισσά σου!



Ας δούμε άλλο ένα παράδειγμα. Έχουμε μια χελωνίτσα που αγαπά την ζωγραφική:

Ποιες οδηγίες πρέπει να της δώσουμε, για να ζωγραφίσει ένα τετράγωνο; Έστω ότι οι διαθέσιμες οδηγίες που μπορούμε να χρησιμοποιήσουμε είναι το σύνολο των κινήσεων που μπορεί να κάνει

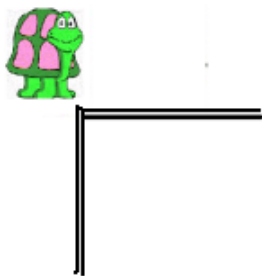
- δεξιά, αριστερά, πάνω, κάτω - και ότι δεν μπορούμε να σηκώσουμε το μολύβι πριν τελειώσει το τετράγωνο.

Οι οδηγίες θα μπορούσαν να είναι οι εξής :

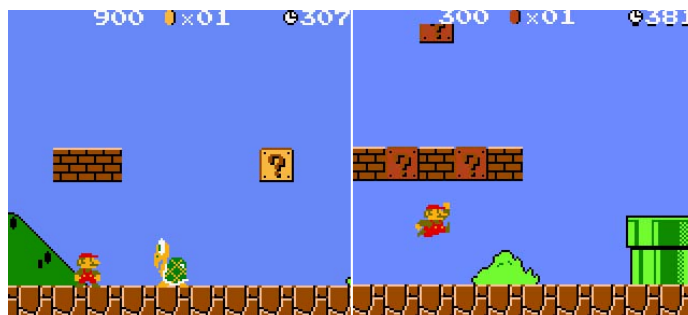
1. Τράβηξε μία γραμμή προς τα δεξιά
2. Τράβηξε μια ίδιου μήκους γραμμή προς τα κάτω
3. Τράβηξε μία ίδιου μήκους γραμμή προς τα αριστερά
4. Τράβηξε μία όμοια γραμμή προς τα πάνω

Τα βήματα που αποτελούν ένα πρόγραμμα λέγονται **εντολές-οδηγίες**. Το κλειδί λοιπόν για την λύση αυτών των προβλημάτων είναι η σωστή διατύπωση των **εντολών-οδηγιών**.

Οι εντολές αυτές πρέπει να ακολουθούν μια λογική σειρά. Σκεφτείτε, τι θα συνέβαινε αν λέγαμε στην χελώνα να κάνει πρώτα το βήμα 3 και έπειτα το βήμα 2. Θα ζωγράφιζε τελικά η χελώνα το τετράγωνο; Προφανώς όχι. Θα προέκυπτε κάτι τέτοιο:



Σκεφτείτε τώρα τι συμβαίνει όταν παίζετε το ηλεκτρονικό παιχνίδι *Super Mario*. Εσείς πατάτε πλήκτρα, ο υπολογιστής αντιστοιχίζει τα πλήκτρα με τις εντολές του προγράμματος και κινεί ανάλογα τον Mario. Πατώντας το δεξί βελάκι, ο Mario προχωράει προς τα δεξιά, με το πλήκτρο κενό (space) χοροπηδάει, με το αριστερό βελάκι γυρίζει πίσω κλπ. Η κίνηση του Mario στον υπολογιστή είναι ένα πρόβλημα που καλείται να λύσει κάποιος προγραμματιστής.



Αυτό που πρέπει να γίνει απολύτως κατανοητό είναι ότι ο υπολογιστής δεν έχει νοημοσύνη. Οι εντολές που χρησιμοποιεί έχουν γραφτεί από κάποιον προγραμματιστή. Έτσι για κάθε πρόβλημα ο υπολογιστής εκτελεί μια σειρά εντολών που θα οδηγήσουν στην επίλυση του.

Φυσικά δεν είναι μόνο τα παιχνίδια, που η υλοποίησή τους βασίζεται πάνω σε αυτή την λογική. Ο διαδικτυακός περιηγητής που σίγουρα έχετε χρησιμοποιήσει για να επισκέπτεστε διάφορες ιστοσελίδες, το πρόγραμμα του ηλεκτρονικού ταχυδρομείου με το οποίο στέλνετε email σε έναν φίλο σας είναι μερικά από τα προγράμματα τα οποία δημιουργήθηκαν από τους προγραμματιστές. Τα προγράμματα αυτά μπορεί να σας φαίνονται πολύ δύσκολο για να δημιουργηθούν, αλλά η λογική που χρησιμοποιήθηκε για να κατασκευαστούν είναι παραπλήσια με τα προβλήματα που αναφέραμε προηγουμένως. Οι προγραμματιστές ανέλυσαν τα προβλήματα σε μικρότερα υποπροβλήματα, έφτιαξαν αλγόριθμους για να βρουν λύσεις σε αυτά και δημιούργησαν τις κατάλληλες ακολουθίες εντολών. Τα προγράμματα είναι μέσα στην ζωή μας ακόμα και αν δεν το καταλαβαίνουμε! Η διαδικασία σύνταξης προγραμμάτων ονομάζεται **προγραμματισμός**.

Ο προγραμματισμός είναι μια ταχύτατα εξελισσόμενη επιστήμη που μέσα από τα διάφορα επιτεύγματα που έχουν επιτύχει οι επαγγελματίες του κλάδου - οι προγραμματιστές - έχει συμβάλει κατά πολύ στην τεχνολογική πρόοδο.

## 1.3 Πως επικοινωνούμε με τους υπολογιστές;

Και ποιες είναι οι οδηγίες-εντολές που καταλαβαίνει ένας υπολογιστής; Πίσω από όλα τα προηγούμενα παραδείγματα κρύβεται μια **γλώσσα προγραμματισμού**, δηλαδή ένας τρόπος με τον οποίο μιλάμε στο υπολογιστή και του διατυπώνουμε μια σειρά εντολών. Προηγουμένως, η γλώσσα προγραμματισμού της χελώνας περιλάμβανε εντολές οι οποίες μας επέτρεπαν να κινήσουμε την χελώνα δεξιά, αριστερά, πάνω, κάτω, ενώ στο παράδειγμα με τον ιππότη, ο ήρωάς μας ακολουθούσε μια παραπλήσια σειρά εντολών.

Διαλέγοντας το τελευταίο παράδειγμα, τι θα γινόταν στην περίπτωση που στη θέση του ιππότη είχαμε έναν υπολογιστή; Με ποιον τρόπο καταλαβαίνει ένας υπολογιστής την σειρά εντολών “Κάνε πέντε βήματα προς τα κάτω”, “Κάνε τέσσερα βήματα προς τα δεξιά” κτλ που δώσαμε στον ιππότη; Και σίγουρα θα προβληματίζεστε για το πως είναι δυνατό να αντιληφθεί μια σειρά βημάτων ένας υπολογιστής αφού δεν έχει ούτε αυτιά για να ακούσει τις εντολές, ούτε μάτια για να δει ένα σχέδιο του λαβυρίνθου με την διαδρομή που θα πρέπει να ακολουθήσει;

Η απάντηση σε όλα τα παραπάνω ερωτήματα είναι πως ναι, ο υπολογιστής μπορεί να καταλάβει μια σειρά ενεργειών που θέλουμε να εκτελέσει. Αρκεί να του μιλήσουμε στην δική του γλώσσα, την μοναδική γλώσσα που καταλαβαίνει και ονομάζεται γλώσσα μηχανής. Ας μάθουμε λοιπόν, κάτι παραπάνω για την γλώσσα μηχανής.

Ανατρέχοντας στην ιστορία των υπολογιστών, βλέπουμε ότι οι πρώτοι υπολογιστές (από την δεκαετία του 1940) δημιουργήθηκαν, για να εκτελούν αριθμητικές πράξεις. Ωστόσο, οι κατασκευαστές της εποχής εκείνης ήρθαν αντιμέτωποι με ένα μεγάλο πρόβλημα. Τα αριθμητικά ψηφία (0, 1, 2, 3, 4, 5, 6, 7, 8 και 9) του δεκαδικού συστήματος ήταν πολλά για να απεικονιστούν με ψηφιακό τρόπο και η κατασκευή ενός τέτοιου υπολογιστή που βασιζόταν σε όλα αυτά τα ψηφία ήταν εξαιρετικά πολύπλοκη. Η λύση ήρθε με την χρήση ενός άλλου συστήματος αρίθμησης, πολύ απλούστερου, του δυαδικού, με το οποίο όλοι οι αριθμοί μπορούσαν να αναπαρασταθούν με τον συνδυασμό δύο καταστάσεων του 0 και 1. Οι δύο αυτές καταστάσεις διευκόλυναν τους κατασκευαστές γιατί τους έδωσε την δυνατότητα να αντιστοιχίσουν:

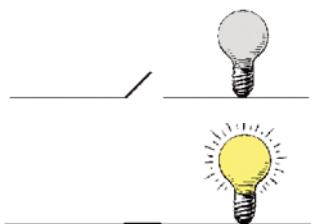
- την απουσία ρεύματος (που περνάει μέσα από τα κυκλώματα του υπολογιστή) με : **0**
- την παρουσία ρεύματος (που περνάει μέσα από τα κυκλώματα του υπολογιστή) με : **1**

και έτσι μπόρεσαν να δημιουργήσουν μια γλώσσα που είχε ένα απλό αλφάβητο με δυο μόνο στοιχεία, το 0 και το 1. Αν ήθελαν λοιπόν, να δώσουν μια απλή εντολή στον υπολογιστή π.χ. να προσθέσει το 5 + 6 και να εμφανίσει το αποτέλεσμα, έπρεπε να μετατρέψουν αυτήν την εντολή, σε μια γραμμή από 0 και 1. Η γλώσσα αυτή ονομάστηκε **γλώσσα μηχανής**. Ήταν όμως εξαιρετικά δύσκολη στη χρήση της! Οι προγραμματιστές σκέφτηκαν ότι ίσως θα μπορούσαν να δημιουργήσουν γλώσσες προγραμματισμού που είναι πιο φιλικές προς αυτούς, δηλαδή γλώσσες που περιέχουν εντολές-οδηγίες που είναι περισσότερο κατανοητές στον ανθρώπινο νου, ενώ στη συνέχεια οι εντολές αυτές θα μπορούσαν να «μεταφράζονται» σε γλώσσα μηχανής.

Έτσι λοιπόν, οι γλώσσες προγραμματισμού λειτουργούν όπως οι φυσικές γλώσσες (Ελληνικά, Αγγλικά, κτλ) που χρησιμοποιούν οι άνθρωποι για να επικοινωνούν μεταξύ τους. Κάθε μία έχει διαφορετικό λεξιλόγιο και συντακτικό αλλά όλες κατασκευάστηκαν για να ικανοποιούν τον ίδιο στόχο, δηλαδή την πιο “ανθρώπινη” επικοινωνία με τον υπολογιστή. Φυσικά τώρα αναρωτιέστε, πόσες γλώσσες υπάρχουν; Μια δεν είναι αρκετή; Ποια η διαφορά της μιας γλώσσας από την άλλη;

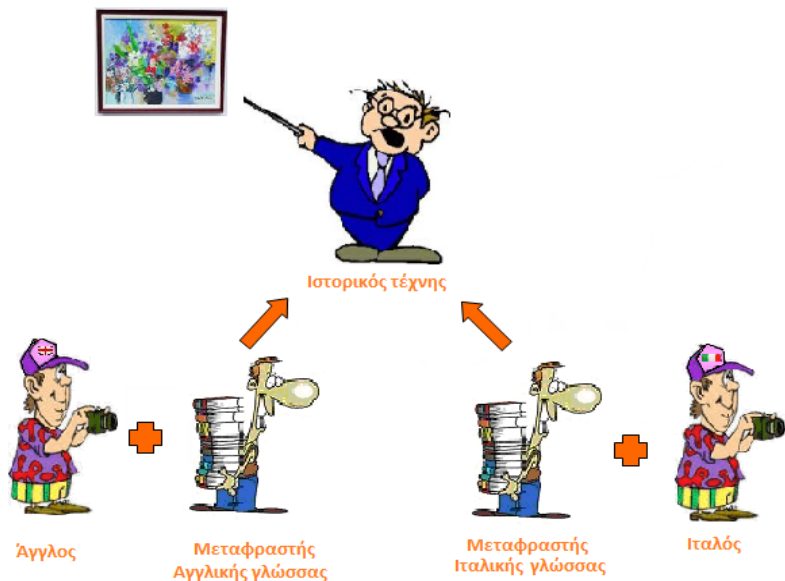
Σε αυτή την φάση δεν χρειάζεται να αγχώνεστε με όλα τα παραπάνω. Αρκεί να αναφέρουμε ότι υπάρχουν πολλές διαφορετικές γλώσσες και ότι ο λόγος ύπαρξής τους είναι ότι καμία δεν μπορεί να ικανοποιήσει τις πολύ διαφορετικές απαιτήσεις των ανθρώπων αλλά και τα ιδιαίτερα χαρακτηριστικά των διαφορετικών προβλημάτων. Κάθε μια από αυτές δημιουργήθηκε για να αντιμετωπίζει διαφορετικά προβλήματα (π.χ. μαθηματικά προβλήματα ή παιχνίδια με γραφικά) και καταλαβαίνετε πως όσο αναπτύσσεται η επιστήμη των υπολογιστών και αυξάνονται οι ανάγκες μας, τόσο θα κάνουν την εμφάνιση τους νέες γλώσσες προγραμματισμού.

Ήδη αναφέραμε πως οι γλώσσες προγραμματισμού αναλαμβάνουν να μετατρέψουν τις εντολές που παρέχουν, σε εντολές γλώσσας μηχανής. Την μετατροπή αυτή εκτελεί ένα πρόγραμμα που



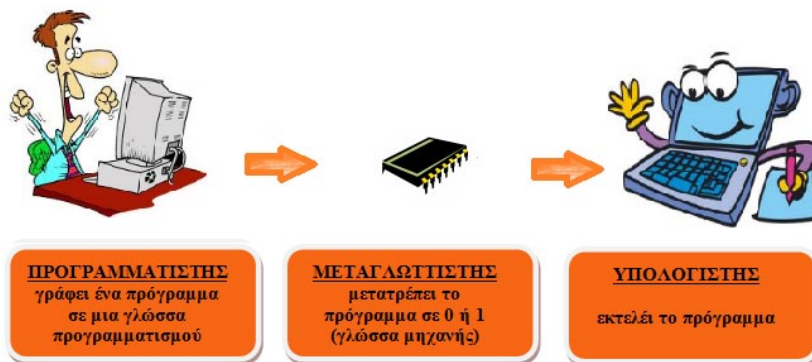
ονομάζεται **μεταγλωττιστής** και την λειτουργία του θα προσπαθήσουμε να εξηγήσουμε με το πιο κάτω παράδειγμα.

Κατά την επίσκεψή σας στην πινακοθήκη της πόλης σας, παρατηρείτε τους διάφορους τουρίστες που έρχονται να θαυμάσουν τα εκθέματα. Στην συνέχεια οι τουρίστες θέλουν να μάθουν περισσότερα για την ιστορία του κάθε εκθέματος και να θέσουν κάποια ερωτήματα στον ιστορικό τέχνης του μουσείου.



Κανένας από αυτούς δεν γνωρίζει την ελληνική γλώσσα, ωστόσο όλα τα γκρουπ των επισκεπτών έχουν μαζί τους από έναν ξεναγό ο οποίος μεταφέρει τα ερωτήματα στον ιστορικό τέχνης. Ο ξεναγός για παράδειγμα, από ένα γκρουπ Άγγλων θα μεταφράσει τα ερωτήματα από τα αγγλικά στα ελληνικά, ο Ιταλός ξεναγός θα μεταφράσει τα ερωτήματα από τα ιταλικά στα ελληνικά κ.ά.

Κατά ανάλογο τρόπο θα μπορούσαμε να πούμε ότι, όπως λειτουργούν οι διαφορετικές φυσικές γλώσσες (Αγγλικά, Ιταλικά, κτλ) που χρησιμοποιούν οι τουρίστες για να επικοινωνούν μεταξύ τους και να διατυπώνουν τα ερωτήματά τους, έτσι λειτουργούν και οι γλώσσες προγραμματισμού για να μπορέσουν οι άνθρωποι να επικοινωνήσουν με τον υπολογιστή. Όπως κάθε γκρουπ έχει τον δικό του ξεναγό, έτσι και κάθε γλώσσα προγραμματισμού έχει τον δικό της μεταγλωττιστή που μετατρέπει τις εντολές της γλώσσας προγραμματισμού σε εντολές της γλώσσας μηχανής. Τέλος, όπως ο ιστορικός τέχνης καταλαβαίνει μόνο την δική του γλώσσα και καμία άλλη, έτσι και ο υπολογιστής καταλαβαίνει μόνο τη γλώσσα μηχανής.



### 1.4 Γιατί να μάθω προγραμματισμό;

Για ποιον λόγο να μάθουμε προγραμματισμό; Έχουμε να κερδίσουμε κάτι ή είναι ένα αντικείμενο με το οποίο ασχολούνται μόνο ειδικοί στους υπολογιστές; Καταρχάς να αναφέρουμε πως εμείς με την βοήθεια του MSKodu θα μάθουμε προγραμματισμό για να κατασκευάζουμε τα δικά μας παιχνίδια! Άρα προβλέπεται πολύ διασκέδαση!

Επιπλέον θα ικανοποιήσετε την περιέργεια που έχετε για τον τρόπο δημιουργίας ενός παιχνιδιού και παράλληλα θα νιώσετε χαρά δημιουργώντας τα δικά σας έργα. Πέρα όμως από το να

κατασκευάζετε παιχνίδια, μπορείτε να σκεφτείτε τον προγραμματισμό σαν ένα λευκό καμβά πάνω στον οποίο θα μπορείτε να ζωγραφίσετε ότι θέλετε. Δηλαδή, θα είσατε σε θέση να κατασκευάσετε ότι προγράμματα θέλετε χωρίς κανείς να περιορίσει τη δημιουργικότητά σας.

Χρησιμοποιώντας γλώσσες προγραμματισμού, θα μπορείτε να φτιάξετε, για παράδειγμα, ένα ηλεκτρονικό ημερολόγιο που θα κρατάει τα γενέθλια και τις γιορτές των φίλων σας και θα σας ενημερώνει όταν πλησιάζει κάθε μία. Θα μπορείτε να φτιάξετε ένα πρόγραμμα στο οποίο να εισάγετε τους βαθμούς των μαθημάτων σας και να σας υπολογίζει το μέσο όρο. Θα μπορούσατε να κατασκευάσετε, ακόμη και έναν ηλεκτρονικό διαχειριστή των οικονομικών σας, με την βοήθεια του οποίου να διαχειρίζεστε καλύτερα το εβδομαδιαίο χαρτζιλί σας. Ο προγραμματισμός λοιπόν μπορεί να διευκολύνει αλλά και να δώσει μια ευχάριστη νότα στην καθημερινότητά σας.

Ακόμη, θα κατανοήσετε πως όλες αυτές οι εφαρμογές με τις οποίες ασχολούμαστε καθημερινά δε δουλεύουν με ένα μαγικό τρόπο, αλλά αξιοποιούν απλές εντολές που καθορίζουν τη συμπεριφορά τους. Θα μπορέσετε για παράδειγμα να καταλάβετε το σκεπτικό ενός προγραμματιστή όταν γράφετε μια έκθεση στον κειμενογράφο, ή όταν ζωγραφίζετε ένα χαμογελαστό πρόσωπο στο πρόγραμμα ζωγραφικής. Κατανοώντας τον προγραμματισμό θα σταματήσετε να αντιμετωπίζετε την τεχνολογία των υπολογιστών ως κάτι ακαταλαβίστικο και ίσως επικίνδυνο. Θα χρησιμοποιείτε τους υπολογιστές με μεγαλύτερη αυτοπεποίθηση, ενώ κατασκευάζοντας τα δικά σας προγράμματα θα γίνετε *συν-δημιουργοί του μέλλοντος*, αφού θα διαπιστώσετε πως εσείς οι ίδιοι μπορείτε να δημιουργείτε τεχνολογία. Μην θεωρείτε πως είμαστε υπερβολικοί! Πολλοί συμμαθητές από διάφορα μέρη του πλανήτη κατάφεραν να κατασκευάσουν καταπληκτικά προγράμματα τα οποία εντυπωσίασαν ακόμα και τους πιο έμπειρους προγραμματιστές.

Φωτεινό παράδειγμα είναι η 12χρονη Sparrow. Το κορίτσι αυτό επιλέχθηκε από την ίδια την Microsoft, την εταιρία που δημιούργησε το MSKodu, για να παρουσιάσει το περιβάλλον του MSKodu στο κοινό. Η Sparrow κατά την επίδειξη δημιούργησε ένα τόσο συναρπαστικό παιχνίδι το οποίο δεν θαύμασε μόνο το κοινό αλλά και πολλοί καταξιωμένοι προγραμματιστές της Microsoft.

Αν θέλετε να δείτε όλη την επίδειξη, με την Sparrow να επιδεικνύει το παιχνίδι μπορείτε, ακολουθώντας τον σύνδεσμο: [Επίδειξη του MSKodu](#). Στην εικόνα φαίνεται η 12χρονη Sparrow με τον προγραμματιστή της Microsoft, Robbie Bach.

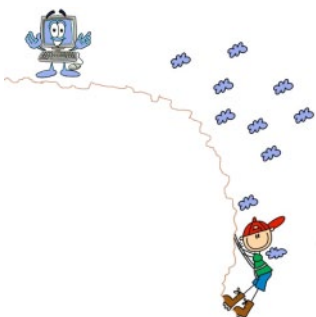
Όμως τα οφέλη από την εκμάθηση του προγραμματισμού δεν σταματούν εδώ. Με τον προγραμματισμό *αναπτύσσετε ακόμη και πνευματικά*. Αρκετές μελέτες έχουν δείξει ότι άτομα τα οποία έχουν μάθει να σκέφτονται με προγραμματιστικό τρόπο, έχουν την ικανότητα να λύνουν τα προβλήματα της καθημερινότητας πιο εύκολα, πιο γρήγορα, πιο αποδοτικά! Μαθαίνοντας να σκέφτεστε με αυτόν τον τρόπο, προβλέπετε καλύτερα τα αποτελέσματα που θα έχει ένας τρόπος επίλυσης ενός προβλήματος αλλά και τις εναλλακτικές επιλογές που πρέπει να ακολουθήσετε, σε περίπτωση που το αρχικό σχέδιο αποτύχει. Θετικές επιδράσεις του προγραμματισμού έχουν βρεθεί ακόμη και στον τομέα των μαθηματικών! Φανταζόμαστε ότι όλοι θέλετε να βελτιώσετε την ταχύτητα και την αποτελεσματικότητα της σκέψης σας.

Τέλος, το συναίσθημα που θα νιώσετε όταν το πρόγραμμά που έχετε δημιουργήσει λειτουργεί και παράγει τα σωστά αποτελέσματα, είναι κάτι που πραγματικά δεν συγκρίνεται! Είναι το συναίσθημα του δημιουργού, είναι το συναίσθημα του ορειβάτη που μόλις κατέκτησε την κορυφή. Θα σας γεμίσει αυτοπεποίθηση και χαρά.

Τι λέτε λοιπόν θέλετε να αρχίσουμε σιγά-σιγά την αναρρίχηση του προγραμματισμού;

## 1.5 Προγραμματισμός και παιχνίδια – Πολλά περιβάλλοντα

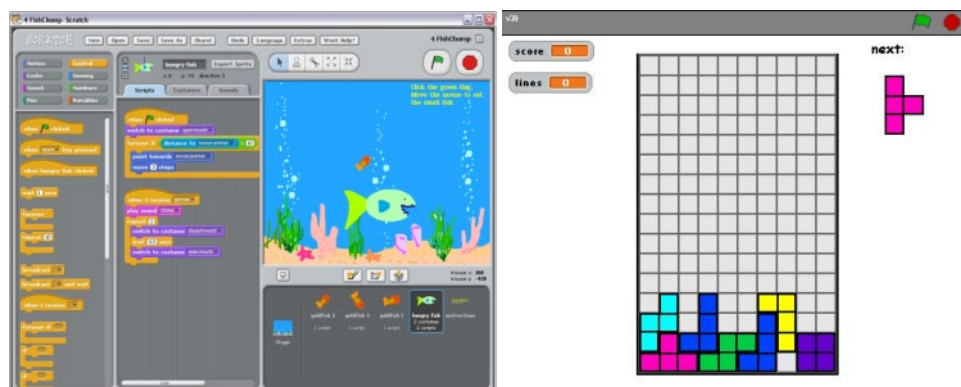
Ήδη έχουμε αναφέρει ότι σκοπός αυτού του βιβλίου είναι να σας μάθει προγραμματισμό δημιουργώντας τα δικά σας παιχνίδια με τη βοήθεια του μικρού ήρωα του βιβλίου, του Kodu.



Εκτός όμως από τον Kodu θα μάθουμε και για άλλους χαρακτήρες, όπως τον *Μηχανάκια*, το *Υποβρύχιο*, το *Χελιδονόψαρο*, το *Κανόνι*, την *Χελώνα*. Θα έχουμε την δυνατότητα να προγραμματίσουμε την συμπεριφορά αυτών των χαρακτήρων επιλέγοντας να κάνουν διάφορες ενέργειες όπως, να πυροβολούν, να τρώνε μήλα, να ακούν ήχους, να κουβαλούν διάφορα αντικείμενα και να τα μετακινούν σε νέες θέσεις.

Το MSKodu δεν είναι το μόνο περιβάλλον με το οποίο μπορούμε να μάθουμε προγραμματισμό με διασκεδαστικό τρόπο. Εμάς μας ενθουσίασε και σας το παρουσιάζουμε. Σας προτρέπουμε όμως να δείτε και άλλα διαφορετικά προγραμματιστικά περιβάλλοντα που μας επιτρέπουν να κατασκευάσουμε τα δικά μας παιχνίδια, να κατασκευάζουμε ρομπότ, να παρουσιάσουμε ιστορίες και να δημιουργούμε ζωγραφιές. Ας δούμε συνοπτικά ορισμένα.

### 1.5.1 Scratch

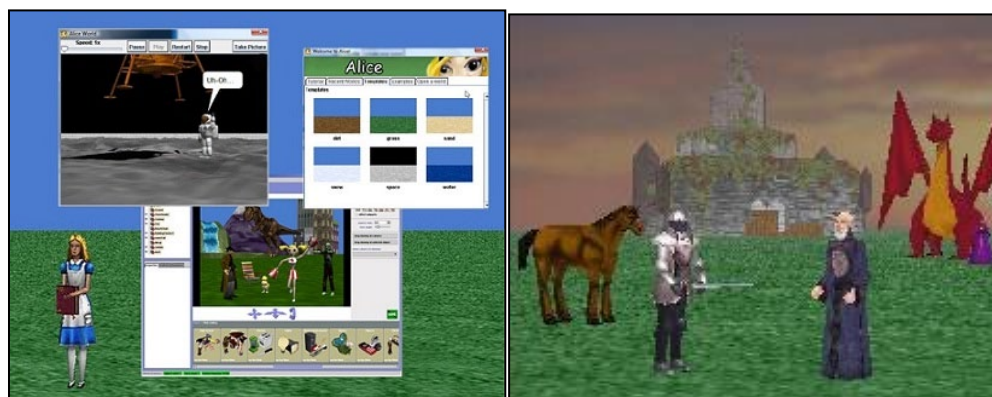


Μέσα από το περιβάλλον του Scratch μπορείτε να φτιάξετε τις δικές σας ιστορίες και παιχνίδια εύκολα και γρήγορα ενώ θα κάνετε παράλληλα μια εισαγωγή στις βασικές έννοιες του προγραμματισμού! Με το Scratch μπορείτε να φτιάξετε το δικό σας pacman, το δικό σας tetris (δεξιά εικόνα) ή τον μοναδικό σας βυθό με μικρά και μεγάλα ψάρια. Με το Scratch δημιουργείτε πολύ εύκολα όλα αυτά τα παιχνίδια μιας και οι εντολές του απεικονίζονται με μικρά τουβλάκια. Αυτά τα τουβλάκια πρέπει να συνθέσετε μεταξύ τους ώστε να δημιουργήσετε όποιο πρόγραμμα έχετε φανταστεί. Κάθε νέο κόσμο που δημιουργείτε έχετε την δυνατότητα να τον βλέπετε άμεσα, να τον τροποποιείτε και να πειραματίζεστε πάνω σε αυτόν.

<http://scratch.mit.edu/><sup>1</sup>

### 1.5.2 Alice 3

Με το περιβάλλον Alice μπορείτε να φτιάξετε εικονικούς κόσμους τριών διαστάσεων μέσα στους οποίους δραστηριοποιούνται αντικείμενα που έχετε δημιουργήσει. Τα αντικείμενα αυτά μπορεί να είναι εξωγήινοι ή διαστημόπλοια, δράκοι ή δεινόσαυροι, ιππότες ή νεράιδες. Αν επιλέξετε το αντικείμενο που θα δημιουργήσετε να είναι άνθρωπος, τότε μπορείτε να του δώσετε χαρακτηριστικά όπως φύλο, χρώμα του δέρματος, ύψος, χρώμα των μαλλιών ή ματιών, να διαλέξετε ρούχα κτλ. Μπορείτε να δημιουργήσετε μια κοπέλα με καστανά μαλλιά και κορμάκι να κάνει πατινάζ σε μια παγωμένη λίμνη, έναν αστροναύτη να προσσεληνώνεται ή να φτιάξετε έναν κόσμο με ιππότες, δράκους και μάγους όπως φαίνεται στις παρακάτω εικόνες:



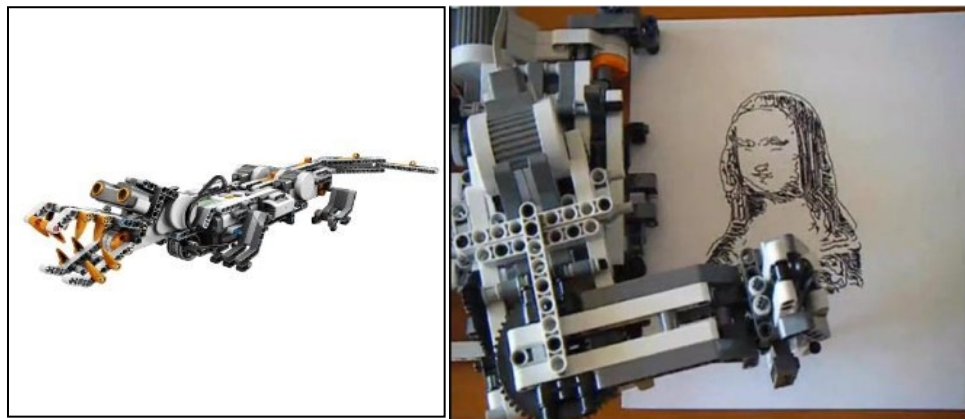
<sup>1</sup> Το αντίστοιχο βιβλίο μας για το Scratch στα ελληνικά, βρίσκεται στη διεύθυνση [www.scratchpay.gr](http://www.scratchpay.gr)

Μπορείτε να δημιουργήσετε ακόμα και το γνωστό παραμύθι του κακού λύκου με τα τρία γουρουνάκια! Παράλληλα με την κατασκευή των κόσμων σας, θα μαθαίνετε προγραμματισμό, αφού έχετε την δυνατότητα να προγραμματίζετε τις ενέργειες που θα κάνουν οι χαρακτήρες σας καθώς και την συμπεριφορά τους όταν έρχονται σε επαφή με άλλους χαρακτήρες.

<http://www.alice.org/>

### 1.5.3 Lego Mindstorms

Τα Lego Mindstorms περιέχουν, τα ήδη γνωστά σε όλους, τουβλάκια Lego τα οποία συναρμολογούμε για να δημιουργήσουμε κάστρα, αυτοκίνητα, πόλεις και ότι άλλο προκύπτει από την φαντασία μας. Όμως με τα Lego Mindstorms δεν έχετε μόνο την δυνατότητα να δημιουργήσετε μια κατασκευή, αλλά μπορείτε να προγραμματίσετε την συμπεριφορά που θέλετε να έχει! Για παράδειγμα μπορείτε να κατασκευάσετε ένα σκυλάκι και να ορίσετε την συμπεριφορά του έτσι ώστε όταν ακούει τη φωνή σας να πλησιάζει προς το μέρος σας ή όταν αντιλαμβάνεται μια κίνηση λίγο πιο μακριά, να ανοίγει το στόμα δείχνοντας απειλητικά τα δόντια του. Εσείς, δηλαδή, προγραμματίζετε τη συμπεριφορά για το ρομποτικό σκυλάκι ώστε να έχει την αίσθηση του ήχου καθώς και την αντίληψη της απόστασης. Και φυσικά το σκυλάκι-ρομποτάκι μπορεί να αποκτήσει και άλλες αισθήσεις αν εσείς το επιθυμείτε. Μπορεί να έχει επιπλέον, την αίσθηση της αφής και του φωτός. Πιο σύνθετα παραδείγματα είναι η κατασκευή ρομπότ-ποδοσφαιριστή που προσπαθεί να βάλει γκολ σε ρομπότ-τερματοφύλακα, ή ρομποτάκια που παλεύουν σε μια μεσαιωνική μάχη για να σώσουν την πριγκίπισσα που βρίσκεται στον πύργο, ή ένα ρομποτάκι, σαν άλλος Leonardo da Vinci, να ζωγραφίζει την Μόνα Λίζα. Και όλα αυτά, προγραμματίζοντας με απλές εντολές.



## 1.6 Kodu και προγραμματισμός

Το MSKodu αρχικά είχε σχεδιαστεί ως εργαλείο μάθησης για τους νέους που χρησιμοποιούν το γνωστό σε εσάς Xbox 360 και κυκλοφόρησε επίσημα στις 7 Ιανουαρίου του 2009. Δύο χρόνια αργότερα, το MSKodu ήταν επιτυχία στο Xbox Live και το χρησιμοποιούσαν σε περισσότερα από 60 εκπαιδευτικά ιδρύματα σε όλη την υφήλιο για να εισάγουν τα παιδιά στον προγραμματισμό. Τώρα όμως είναι διαθέσιμο δωρεάν και για το PC.

Χρησιμοποιώντας τον ήρωα Kodu, μπορείτε να δημιουργήσετε απίθανα παιχνίδια και να προκαλέσετε τους γύρω σας να τα τερματίσουν!

Ας κάνουμε και μια βόλτα στα περιεχόμενα του βιβλίου.

Στο κεφάλαιο 2 θα μάθετε πώς να εγκαταστήσετε το MSKodu και θα παίξετε τα πρώτα παιχνίδια σας! Στο κεφάλαιο 3 ξεκινάτε από τόσο νωρίς να φτιάχνετε τα δικά σας παιχνίδια. Στο κεφάλαιο 4 θα μάθετε να δημιουργείτε τους δικούς σας εντυπωσιακούς. Στα κεφάλαια 5, 6 και 7 θα μελετήσουμε πως δημιουργούμε αντικείμενα, πως προγραμματίζουμε τη συμπεριφορά τους, πως τους επιτρέπουμε να αλληλεπιδρούν με τον χρήστη, με άλλα αντικείμενα και τον κόσμο του παιχνιδιού μας. Στο κεφάλαιο 8 θα σας δείξουμε πως τα παιχνίδια σας μπορούν να αποκτήσουν στοιχεία όπως ζωές, σκορ, χρόνο,





ενέργεια. Στο κεφάλαιο 9 θα συζητήσουμε κάποιες πιο σύνθετες προγραμματιστικές τεχνικές ώστε τα παιχνίδια σας να γίνουν πιο αληθοφανή! Στο κεφάλαιο 10, θα κάνουμε ένα μικρό διάλλειμα και θα σας παρουσιάσουμε βασικές αρχές της μηχανικής παιχνιδιών, δηλαδή των σχεδιαστικών επιλογών που πρέπει να λαμβάνουμε υπόψη ώστε να δημιουργούμε προκλητικά παιχνίδια. Τέλος, στα κεφάλαια 11 και 12 θα εφαρμόσουμε όλες τις γνώσεις των προηγούμενων κεφαλαίων για να δημιουργήσουμε μαζί 2 ολοκληρωμένα παιχνίδια. Θυμηθείτε ότι ο μόνος περιορισμός στον προγραμματισμό είναι η φαντασία σας!

## Περίληψη

Στο κεφάλαιο αυτό συζητήσαμε για την έννοια του προγραμματισμού, η οποία όπως αποδείχθηκε δεν είναι τελείως καινούρια καθώς, είτε το αντιλαμβανόμαστε είτε όχι, υπάρχει στην ζωή μας! Μέσα από παραδείγματα, είδαμε πως η καθημερινότητά μας αποτελείται από αλγορίθμους που δημιουργούμε για να λύσουμε τα προβλήματά μας, ενώ κατανοήσαμε περισσότερο την έννοια του προγράμματος τόσο στην ζωή μας όσο και σε σχέση με τους υπολογιστές. Στη συνέχεια μπορέσαμε να δούμε με ποιον τρόπο καταφέρνουμε να δίνουμε οδηγίες στον υπολογιστή και μιλήσαμε για τις γλώσσες προγραμματισμού. Επιπλέον πήραμε απαντήσεις για το τι όφελος έχουμε μαθαίνοντας προγραμματισμό και είδαμε πως στο βιβλίο αυτό, με την βοήθεια του ήρωά μας, του Kodu, θα μάθουμε προγραμματισμό φτιάχνοντας τα δικά μας παιχνίδια, δηλαδή, διασκεδάζοντας! Συζητήσαμε για εναλλακτικά προγραμματιστικά περιβάλλοντα με παρόμοιο χαρακτήρα με το MSKodu και τέλος ρίξαμε μια κλεφτή ματιά στα πιο σημαντικά στοιχεία που θα αναλυθούν στην συνέχεια. Σας προτείνουμε, λοιπόν να διαβάσετε τα ακόλουθα κεφάλαια, να γνωρίσετε τις μαγικές δυνατότητες του Microsoft Kodu και να επιδοθείτε στην διερεύνηση της δημιουργικότητάς σας! Καλή μελέτη!

## Ερωτήσεις

1. Περιγράψτε τι είναι ένα πρόγραμμα και ποια είναι τα βασικά του χαρακτηριστικά.
2. Ποιες είναι οι διαφορές μεταξύ του αλγορίθμου και του προγράμματος;
3. Ποια πιστεύετε ότι είναι τα οφέλη που μπορεί να σας προσφέρει ο προγραμματισμός στην καθημερινή σας ζωή;
4. Περιγράψτε καθημερινές σας δραστηριότητες στις οποίες χρησιμοποιείτε προγραμματιστικές εφαρμογές.
5. Ποιος πιστεύετε ότι είναι ο λόγος που δημιουργήθηκαν οι γλώσσες προγραμματισμού; Τι κοινό έχουν μεταξύ τους; Αρκεί μία γλώσσα προγραμματισμού;

## Δραστηριότητες

1. Περιγράψτε τον αλγόριθμο 7 βημάτων, που θα δίνετε στην μητέρα σας για να φτιάξει το αγαπημένο σας κέικ!
  - Στις οδηγίες που θα δώσετε, χρησιμοποιήστε τα παρακάτω: αλεύρι, μείγμα, αυγά, ζάχαρη, ρίξε, ανακάτεψε, βγάλε, σοκολάτα, φόρμα ψησίματος, βάλε, φούρνος, άφησε να ψηθεί 1 ώρα.
  - Στην συνέχεια συγκρίνετε τον αλγόριθμό σας με τους αλγορίθμους που κατασκεύασαν οι υπόλοιποι συμμαθητές σας. Τι παρατηρείτε και σε ποιο συμπέρασμα καταλήγετε;
2. Ανακαλύψτε τις κρυμμένες λέξεις: προγραμματισμός, μεταγλώττιση, αλγόριθμος, γλώσσα, εκτέλεση, εντολή παιχνίδι.

Π	Ω	Α	Ε	Τ	Υ	Γ	Λ	Ω	Σ	Σ	Α	Η	Ξ	Φ
Σ	Ρ	Δ	Ω	Μ	Δ	Ι	Ο	Π	Ρ	Φ	Ε	Ι	Χ	Β
Ψ	Ω	Ο	Α	Φ	Μ	Λ	Φ	Τ	Θ	Ρ	Δ	Κ	Ν	Μ
Σ	Φ	Ζ	Γ	Ψ	Ε	Κ	Τ	Ε	Λ	Ε	Σ	Η	Λ	Ρ
Ε	Ρ	Ν	Υ	Ρ	Τ	Θ	Χ	Ν	Μ	Δ	Τ	Φ	Σ	Α
Θ	Ε	Δ	Λ	Β	Α	Λ	Γ	Ο	Ρ	Ι	Θ	Μ	Ο	Σ
Γ	Ν	Θ	Α	Ξ	Γ	Μ	Ι	Υ	Υ	Θ	Η	Ι	Ρ	Ν
Δ	Τ	Η	Σ	Γ	Λ	Φ	Μ	Τ	Γ	Κ	Α	Ο	Τ	Κ
Β	Ο	Ξ	Δ	Κ	Ω	Ω	Π	Α	Ι	Χ	Ν	Ι	Δ	Ι
Ε	Λ	Φ	Γ	Η	Τ	Σ	Ε	Δ	Τ	Σ	Φ	Ε	Κ	Α
Υ	Η	Γ	Ξ	Β	Τ	Ο	Γ	Ζ	Κ	Ι	Λ	Α	Δ	Σ
Θ	Μ	Λ	Τ	Σ	Ι	Υ	Ξ	Η	Α	Μ	Σ	Σ	Γ	Ε
Π	Κ	Ε	Φ	Τ	Σ	Β	Φ	Ρ	Ν	Α	Ζ	Μ	Η	Θ
Η	Λ	Σ	Ε	Θ	Η	Ρ	Σ	Μ	Π	Ρ	Φ	Π	Ο	Ι
Ν	Τ	Ξ	Υ	Κ	Τ	Ι	Τ	Α	Θ	Μ	Η	Δ	Μ	Σ